Report on Particle Swarm Optimization

Surya Raghav, Vibulan, Srinidhi, Srinivas
CS21B2042, CS21B2043, CS21B2044, CS21B2045

Dec 4, 2022

1 Introduction

The particle swarm optimization (PSO) algorithm is nature inspired meta-heureitic that is used to to solve optimization problems. We study the PSO algorithm used to find the global optimum (minimum/ maximum) of an objective function.

PSO is a stochastic optimization technique based on swarm of animals, which was proposed by Eberhart and Kennedy (1995). PSO algorithm simulates social behavior of animals such as insects, herds, birds, and fishes. These swarms (group of individuals) conform to a cooperative way to find food, and each member in the swarms keeps changing the search pattern according to the learning experiences of its own and other members.

The main design idea of the PSO algorithm is closely related to two research areas:

1. Evolutionary Computation: evolutionary computation is a family of algorithms for global optimization inspired by biological evolution. It is a population based meta-heuristic. It uses a swarm of individuals (solutions) which makes it possible to simultaneously search large regions in the solution space of the objective function.

2. Artificial Life: It studies the behavior of social animals from the viewpoint of artificial life theory, i.e, how to construct swarm of artificial life systems with cooperative behavior that mimics real life.

- Swarm Intelligence (SI): It is the collective behavior of decentralized, self-organized systems, natural or artificial. SI systems consist typically of a population of simple agents interacting locally with one another and with their environment. The inspiration often comes from nature. It draws ideas from Evolutionary Computation and Artificial Life.

2 Motivation

Optimization is very important in many applications such as from engineering design, data mining and machine learning. However, traditional gradient based methods such as gradient descent cannot be used to find the global optimum as they converge at local optimum. Hence, they cannot be used for functions such as the Rastrigin function with multiple local minima and maxima. Moreover many real world data do not have gradient information available. Despite the increasing capacity of computers, simple brute force strategies are not practical. PSO algorithm overcomes these difficulties. It is an efficient algorithm that finds the optimum of an objective function. It makes no assumptions about the problem being optimized and can search very large spaces. It requires less tuning and converges quickly in a large number of functions. It is also less dependent on the set of initial points as compared to other optimization techniques.

3 Background

The development of the PSO algorithm is preceded by an early simple model of swarm intelligence, namely Boid (Bird-oid) model (Reynolds 1987). This model is designed to simulate the behavior of birds.

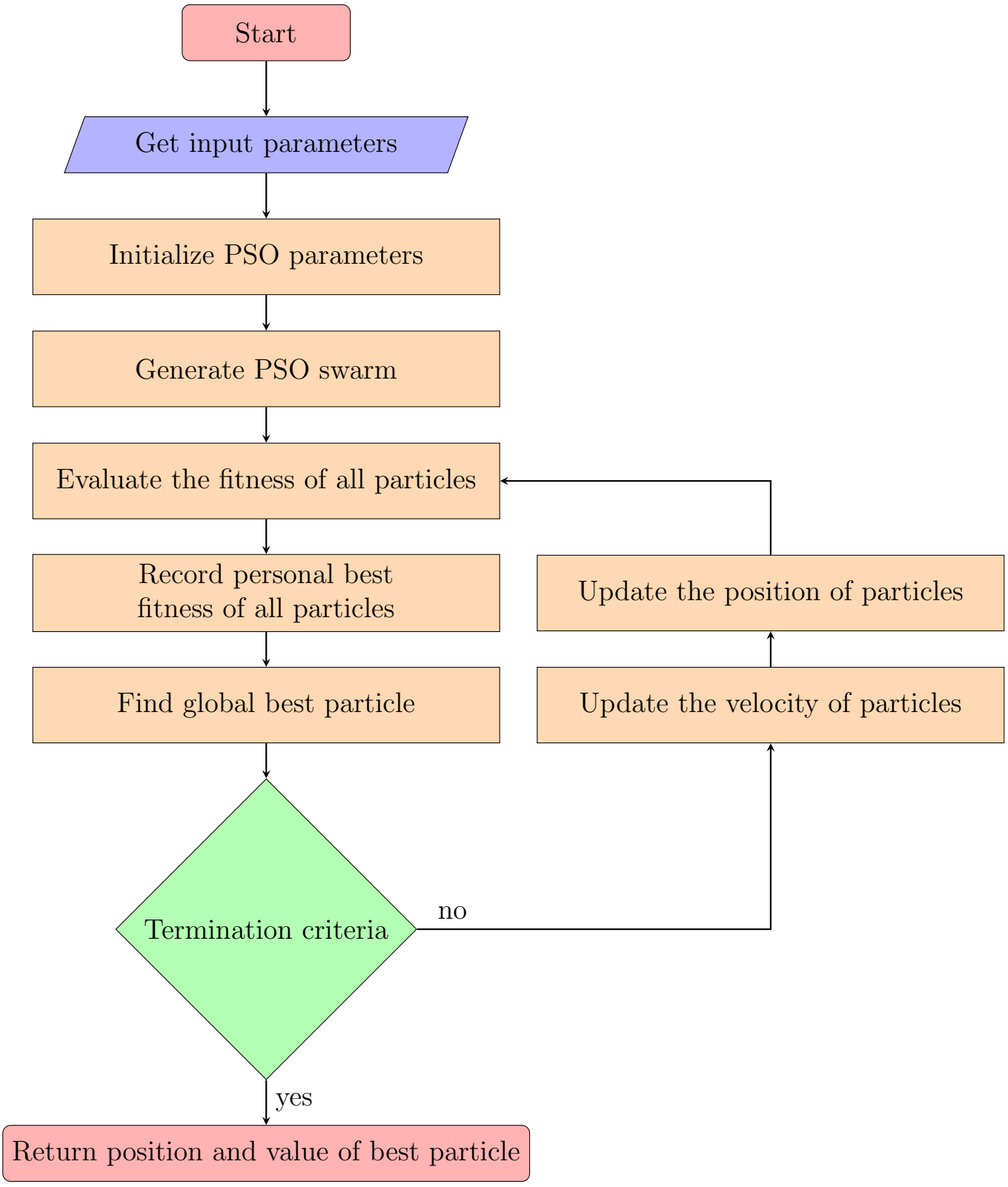
The simplest model can be depicted as follows. Each individual of the birds is represented by a point in the Cartesian coordinate system, randomly assigned with initial velocity and position. Then run the program in accordance with "the nearest proximity velocity match rule," so that one individual has the same speed as its nearest neighbor. With the iteration going on in the same way, all the points will have the same velocity quickly. As this model is too simple and far away from the real cases, a random variable is added to the speed item. That is to say, at each iteration, aside from meeting "the nearest proximity velocity match," each speed will be added with a random variable, which makes the total simulation to approach the real case.

The PSO algorithm uses swarm intelligence to find global optimum of a function.

4 Idea

Moving particles in the search space, which improves their fitness by interaction with other particles.

4.1 Flowchart of Algorithm



Algorithm - Parameters:

- f: objective function
- $\bar{x}_i$: position of the particle
- $\bar{v}_i$: position of the particle

- ◇ A: Population of particles
- ◇ w: Inertia weight
- ◇ $c1$: Cognitive constant
- ◇ $c2$: Social constant
- ◇ $r1, r2$: Random number $\in$ [0, 1]

4.2 Algorithm Steps

1. Create a 'population' of agents (particles) uniformly distributed over X, where X is the Domain or the search region.
2. Evaluate each particle's position according to the objective function.
3. If a particle's current position is better than its previous best position, (For minimization problems if the value of objective function is lesser than the value at its previous best position, For maximization problems if the value of objective function is greater than the value at its previous best position), update best position.
4. Determine the best particle (according to the particle's previous best positions).
5. Update particles' velocities:

$$\bar{v}_i^{(t+1)} = w\bar{v}_i^{(t)} + c_1 r_1(pbest_i^{(t)} - \bar{x}_i^{(t)}) + c_2 r_2(gbest^{(t)} - \bar{x}_i^{(t)})$$

where $v_i^{(t)}$ is the velocity of particle i in iteration t, $x_i^{(t)}$ is the position of particle i in iteration t, $pbest_i^{(t)}$ is the best position of particle i so far, $gbest^{(t)}$ is the best position found so far over all particles.

   (a) $w\bar{v}_i^{(t)}$ is the **Inertia** component: Makes the particle move in the same direction and with the same velocity

   (b) $c_1 r_1(pbest_i^{(t)} - \bar{x}_i^{(t)})$ is the **Personal** component: Makes the particle return to a previous position, better than the current position. This is based on real life observation that individuals stick to the old way that have proven successful in the past and resist new changes. In PSO the particle remembers its best position in the past, and it would like to change its velocity to return to that position.

(c) $c_2 r_2(gbest^{(t)} - \bar{x}_i^{(t)})$ is the **Social** component: Makes the particle follow the best neighbours direction. This is based on the real life observation - "If it worked for them, then maybe it will work for me too." In PSO this corresponds to the particles moving towards the best position.

The velocity update equation can also be interpreted as:

(a) $w\bar{v}_i^{(t)}$ is the **Diversification** component: Searches new solutions and finds the regions with potentially the best solutions

(b) $c_1 r_1(pbest_i^{(t)} - \bar{x}_i^{(t)}) + c_2 r_2(gbest^{(t)} - \bar{x}_i^{(t)})$ is the **Intensification** component: Explores the previous solutions and finds the best solution for a given region

6. Move particles to their new positions:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}$$

7. Go to step 2 until termination criteria is reached. Termination criteria is usually a fixed number of iterations.

5 Implementation and Analysis

The PSO algorithm is implemented (see submitted code) in python and tested for various functions.

Example

Minimize the the rastrigin function in 2 dimensions:

$$f(x_1,\, x_2) = 20 + (x_1^2 - 10cos(2\pi x_1)) + (x_2^2 - 10cos(2\pi x_2))$$

$x_1, x_2 \in$ [-5, 5]
The rastrigin function has multiple local minima, hence gradient based methods cannot be used to globally optimize the function. PSO algorithm works very well for this function.

The function is optimized using the PSO algorithm implemented in python. Using 20 particles, PSO algorithm is run for 50 iterations. The movement of the particle with best fitness is plotted. The value of function at *gbest* at every iteration is also plotted.

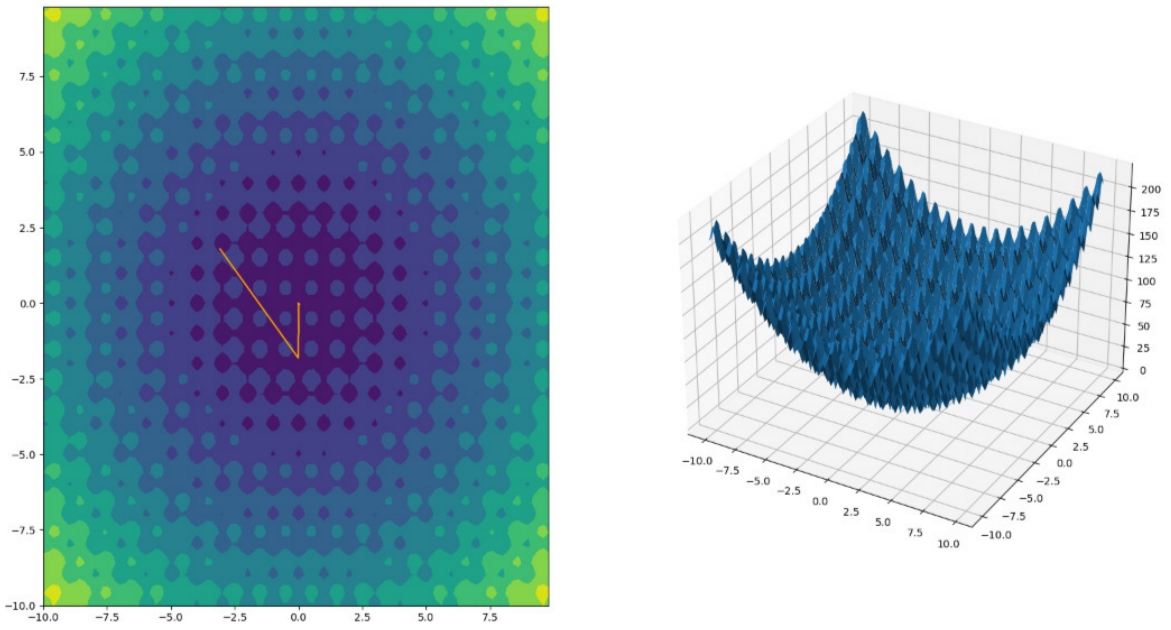


Figure 1: Movement of *gbest* over the contour plot and 3D plot of rastrigin function

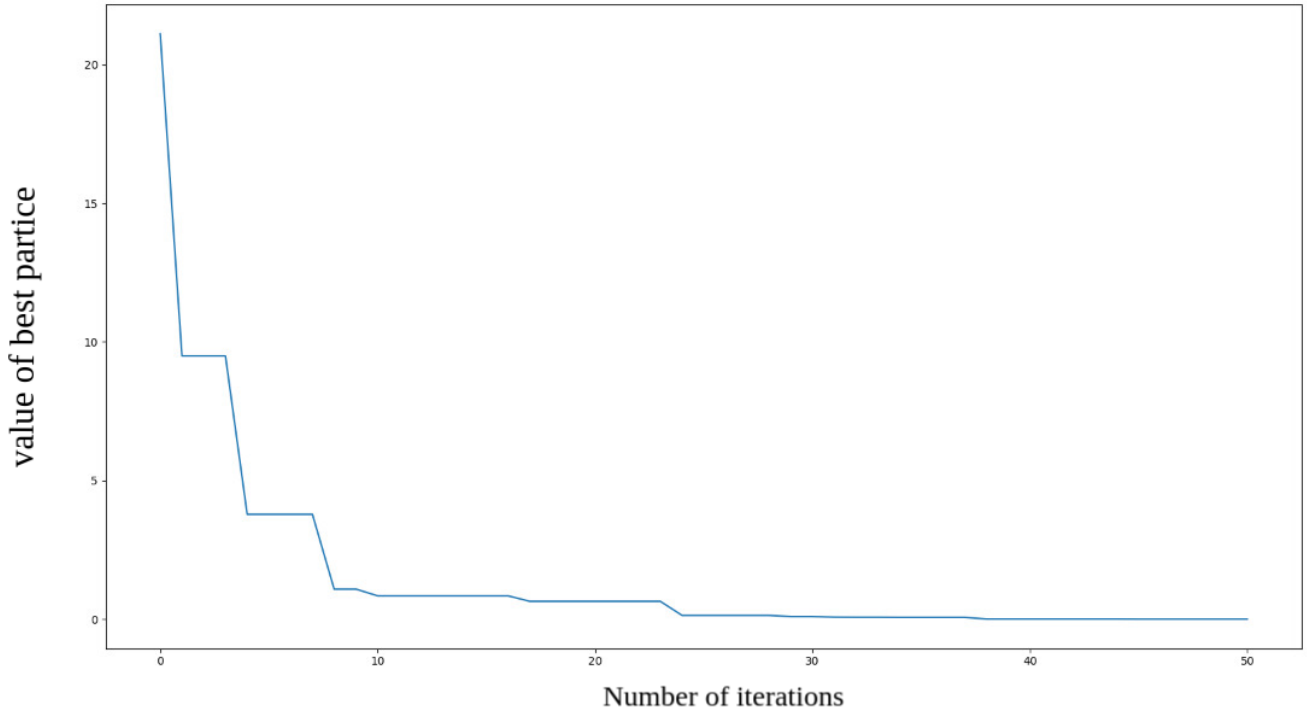


Figure 2: Iteration vs. $f(gbest^{(t)})$

The position of points are plotted as black dots on the contour plot of rastrigin function over various iterations. The particles converge to the global optimum.

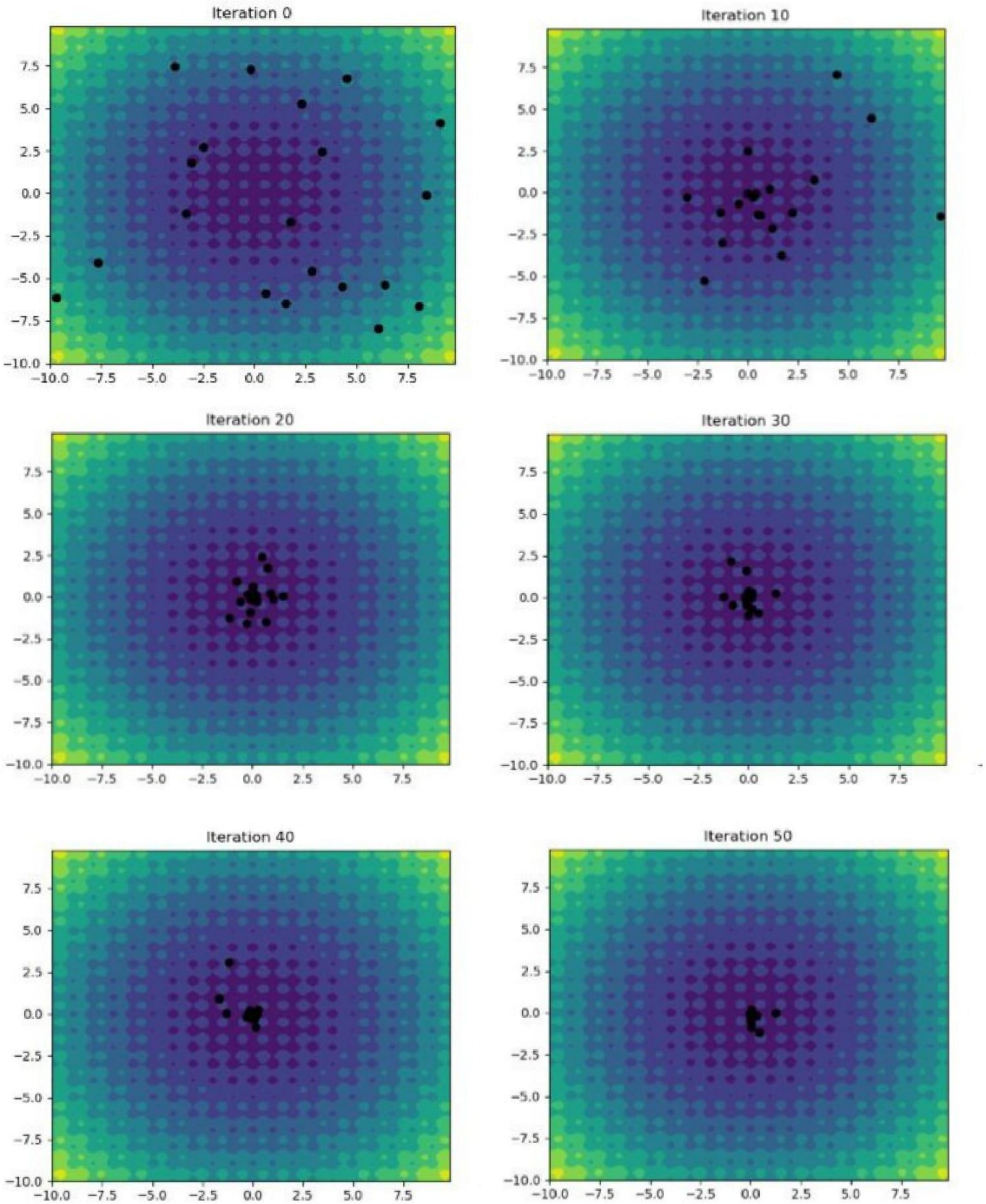


Figure 3: Scatter plot of particles over the contour plot at various iterations

Particle values at iterations

The particle values are shown at first, second and last iteration.

x1	x2	v1	v2	pb1	pb2	Best	Fitness	Gbest
-6.456	0.127	1.187	0.491	-6.456	0.127	64.333	64.333	10.408
-2.171	-5.750	-5.600	-8.495	-2.171	-5.750	52.983	52.983	10.408
-8.006	-3.727	-4.047	-3.349	-8.006	-3.727	89.427	89.427	10.408
-1.284	2.874	6.490	-4.108	-1.284	2.874	24.987	24.987	10.408
2.158	0.079	-2.030	-0.813	2.158	0.079	10.408	10.408	10.408
3.912	9.819	4.735	3.308	3.912	9.819	118.988	118.988	10.408
-8.015	-2.938	-5.286	-5.942	-8.015	-2.938	73.653	73.653	10.408
-8.659	7.430	1.353	-5.454	-8.659	7.430	164.634	164.634	10.408
6.259	0.852	-1.965	0.879	6.259	0.852	54.490	54.490	10.408
-7.738	-1.623	-4.564	9.194	-7.738	-1.623	90.412	90.412	10.408
-5.377	-3.868	-0.336	-0.843	-5.377	-3.868	64.278	64.278	10.408
9.098	0.397	-9.028	8.259	9.098	0.397	102.720	102.720	10.408
1.212	0.230	7.214	-9.150	1.212	0.230	17.901	17.901	10.408
5.765	8.912	-8.339	4.348	5.765	8.912	123.193	123.193	10.408
-4.942	0.651	9.524	-8.988	-4.942	0.651	41.335	41.335	10.408
-4.228	8.013	6.033	3.563	-4.228	8.013	90.733	90.733	10.408
4.801	-3.035	-8.948	4.104	4.801	-3.035	39.328	39.328	10.408
-0.086	-7.570	-0.108	-5.758	-0.086	-7.570	77.800	77.800	10.408
-1.576	-5.991	6.419	-5.250	-1.576	-5.991	57.264	57.264	10.408
-3.827	0.563	-4.499	7.766	-3.827	0.563	39.516	39.516	10.408

x1	x2	v1	v2	pb1	pb2	Best	Fitness	Gbest
-5.269	0.418	1.187	0.291	-5.269	0.418	57.811	57.811	10.408
-4.964	-5.750	-2.794	-4.457	-2.171	-5.750	52.983	67.951	10.408
-2.199	-2.889	5.807	0.838	-2.199	-2.889	22.373	22.373	10.408
6.864	-2.895	8.148	-5.769	-1.284	2.874	24.987	61.029	10.408
0.678	-0.514	-1.480	-0.593	2.158	0.079	10.408	35.043	10.408
6.983	9.819	3.071	0.297	3.912	9.819	118.988	151.017	10.408
-8.015	-7.065	-3.163	-4.127	-8.015	-2.938	73.653	115.005	10.408
0.225	-1.913	8.884	-9.343	0.225	-1.913	13.579	13.579	10.408
0.779	0.729	-5.480	-0.122	0.779	0.729	20.625	20.625	10.408
-0.970	6.816	6.768	8.439	-0.970	6.816	53.563	53.563	10.408
-5.714	1.227	-0.336	5.095	-5.714	1.227	54.946	54.946	10.408
0.069	6.058	-9.028	5.661	0.069	6.058	38.287	38.287	10.408
6.537	-6.451	5.325	-6.681	1.212	0.230	17.901	123.603	10.408
-2.573	0.033	-8.339	-8.879	-2.573	0.033	25.795	25.795	10.408
4.582	-6.261	9.524	-6.912	-4.942	0.651	41.335	89.583	10.408

x1	x2	v1	v2	pb1	pb2	Best	Fitness	Gbest
0.145	0.509	3.519	-0.075	0.145	0.509	24.139	24.139	3.836
1.944	-2.493	6.115	5.061	1.944	-2.493	30.604	30.604	3.836
7.630	1.929	2.258	1.934	-2.199	-2.889	22.373	79.766	3.836
-3.496	4.090	-7.370	5.325	-1.284	2.874	24.987	50.498	3.836
3.104	0.458	2.467	0.988	2.158	0.079	10.408	31.568	3.836
-0.675	-1.042	-5.017	-5.431	-0.675	-1.042	16.415	16.415	3.836
7.474	6.829	8.342	4.528	-0.868	2.301	22.448	127.592	3.836
-0.181	0.499	-8.451	7.605	0.225	-1.913	13.579	26.073	3.836
-4.289	0.057	-1.707	-0.285	0.779	0.729	20.625	31.481	3.836
8.406	0.940	4.915	-2.461	-0.970	6.816	53.563	90.548	3.836
1.117	-1.114	4.573	-5.690	1.117	-1.114	7.552	7.552	3.836
-2.793	6.807	3.358	-2.341	0.069	6.058	38.287	67.963	3.836
-0.697	3.533	-3.955	4.887	1.212	0.230	17.901	46.050	3.836
2.310	-1.875	7.463	4.531	2.310	-1.875	25.468	25.468	3.836
-4.647	-6.261	-1.422	-6.912	-4.942	0.651	41.335	87.497	3.836
1.080	-0.948	-6.565	-1.309	1.080	-0.948	3.836	3.836	3.836
-5.372	6.314	-0.155	1.212	4.801	-3.035	39.328	99.571	3.836
2.800	1.356	0.099	4.133	2.800	1.356	32.761	32.761	3.836
-0.872	-4.115	-8.165	-7.647	-0.872	-4.115	23.268	23.268	3.836
5.197	-4.118	1.477	-5.933	3.720	1.816	34.975	53.335	3.836

x1	x2	v1	v2	pb1	pb2	Best	Fitness	Gbest
4.032	-1.605	3.887	-2.114	0.145	0.509	24.139	36.955	3.836
5.786	2.297	3.842	4.790	1.944	-2.493	30.604	59.434	3.836
1.010	-0.668	-6.620	-2.598	1.010	-0.668	16.389	16.389	3.836
-5.032	5.180	-1.536	1.090	-1.284	2.874	24.987	58.103	3.836
1.013	-1.156	-2.091	-1.614	1.013	-1.156	6.811	6.811	3.836
-3.132	-4.937	-2.457	-3.895	-0.675	-1.042	16.415	38.199	3.836
8.739	6.060	1.266	-0.769	-0.868	2.301	22.448	124.477	3.836
-5.207	3.604	-5.026	3.106	0.225	-1.913	13.579	65.386	3.836
-1.431	-0.900	2.858	-0.956	0.779	0.729	20.625	23.862	3.836
4.005	2.413	-4.400	1.474	4.005	2.413	40.430	40.430	3.836
4.414	-5.099	3.297	-3.984	1.117	-1.114	7.552	65.938	3.836
3.717	-0.208	6.509	-7.015	3.717	-0.208	33.338	33.338	3.836
-1.746	2.520	-1.050	-1.013	1.212	0.230	17.901	39.554	3.836
6.279	2.537	3.969	4.412	2.310	-1.875	25.468	77.432	3.836
-0.283	-5.938	4.365	0.323	-4.942	0.651	41.335	48.125	3.836
-3.706	-1.903	-4.786	-0.954	1.080	-0.948	3.836	31.919	3.836
4.212	-2.305	9.583	-8.620	4.801	-3.035	39.328	44.059	3.836

```
     0.163    -0.028    -0.348     0.016    -0.955     0.024     1.423     4.996     0.000
     0.105     0.015    -0.057    -0.009    -0.012    -0.005     0.032     2.164     0.000
    -0.001    -0.002     0.016     0.010    -0.001    -0.002     0.001     0.001     0.000
    -4.234     1.038    -5.221     1.574    -0.970     0.953     2.466    28.312     0.000
     0.007     0.152     0.011    -0.108     0.031    -0.022     0.286     4.283     0.000
    -0.211     1.188    -0.032     0.256    -0.055     0.152     4.837    15.271     0.000
     0.244    -0.022     1.010    -0.007    -0.064    -0.029     0.972     9.762     0.000
     0.017     0.001     0.033     0.003     0.002    -0.002     0.002     0.060     0.000
     0.001     0.000     0.000    -0.000     0.000     0.000     0.000     0.000     0.000
     0.470     0.034     0.408     0.010     0.063     0.024     0.883    20.279     0.000
     0.377    -0.017    -2.534     0.190     1.924    -0.128     7.894    17.348     0.000
    -0.001     0.032    -0.003     0.071    -0.000     0.004     0.003     0.207     0.000
    -0.175     0.003    -0.122     0.002    -0.040     0.000     0.315     5.497     0.000
     0.008    -0.008    -0.004     0.004     0.003    -0.003     0.003     0.028     0.000


       x1        x2        v1        v2        pb1       pb2      Best   Fitness     Gbest
     0.921    -0.541     3.167    -1.860    -1.853     1.086    10.019    22.014     0.000
    -0.049    -0.025    -0.001    -0.006    -0.003    -0.014     0.041     0.606     0.000
    -0.163     0.177    -0.317     0.352    -0.059     0.059     1.353    10.426     0.000
     0.015    -0.018    -0.037     0.006     0.015    -0.018     0.114     0.114     0.000
     0.016    -0.004     0.062     0.017     0.014    -0.009     0.053     0.054     0.000
     0.002     0.036     0.000     0.077     0.004    -0.007     0.014     0.258     0.000
    -1.102     0.051    -1.265     0.079    -0.955     0.024     1.423     3.698     0.000
    -0.115    -0.019    -0.221    -0.033    -0.012    -0.005     0.032     2.593     0.000
     0.011     0.006     0.012     0.008    -0.001    -0.002     0.001     0.032     0.000
    -4.903     1.940    -0.669     0.902    -0.970     0.953     2.466    30.290     0.000
     0.012    -0.153     0.005    -0.306     0.031    -0.022     0.286     4.352     0.000
     0.189    -1.203     0.400    -2.392    -0.055     0.152     4.837    14.828     0.000
     0.621     0.002     0.377     0.024    -0.064    -0.029     0.972    17.646     0.000
     0.004    -0.001    -0.013    -0.002     0.002    -0.002     0.002     0.004     0.000
     0.000     0.000    -0.000     0.000     0.000     0.000     0.000     0.000     0.000
     0.045     0.001    -0.425    -0.034     0.045     0.001     0.407     0.407     0.000
    -0.326     0.038    -0.703     0.054     1.924    -0.128     7.894    15.007     0.000
    -0.002     0.054    -0.001     0.022    -0.000     0.004     0.003     0.577     0.000
     0.069    -0.002     0.245    -0.005    -0.040     0.000     0.315     0.942     0.000
    -0.011     0.011    -0.019     0.019     0.003    -0.003     0.003     0.045     0.000


Best value : 0.0000124175029990918
Best position [0.00020425 0.00014448]
[surya@arch OTML]$
```

6 Disadvantages

PSO algorithm was used to optimize the rastrigin function again with 5 particles and 20 iterations. The program was run thrice and the value of *gbest* and its movement was plotted.

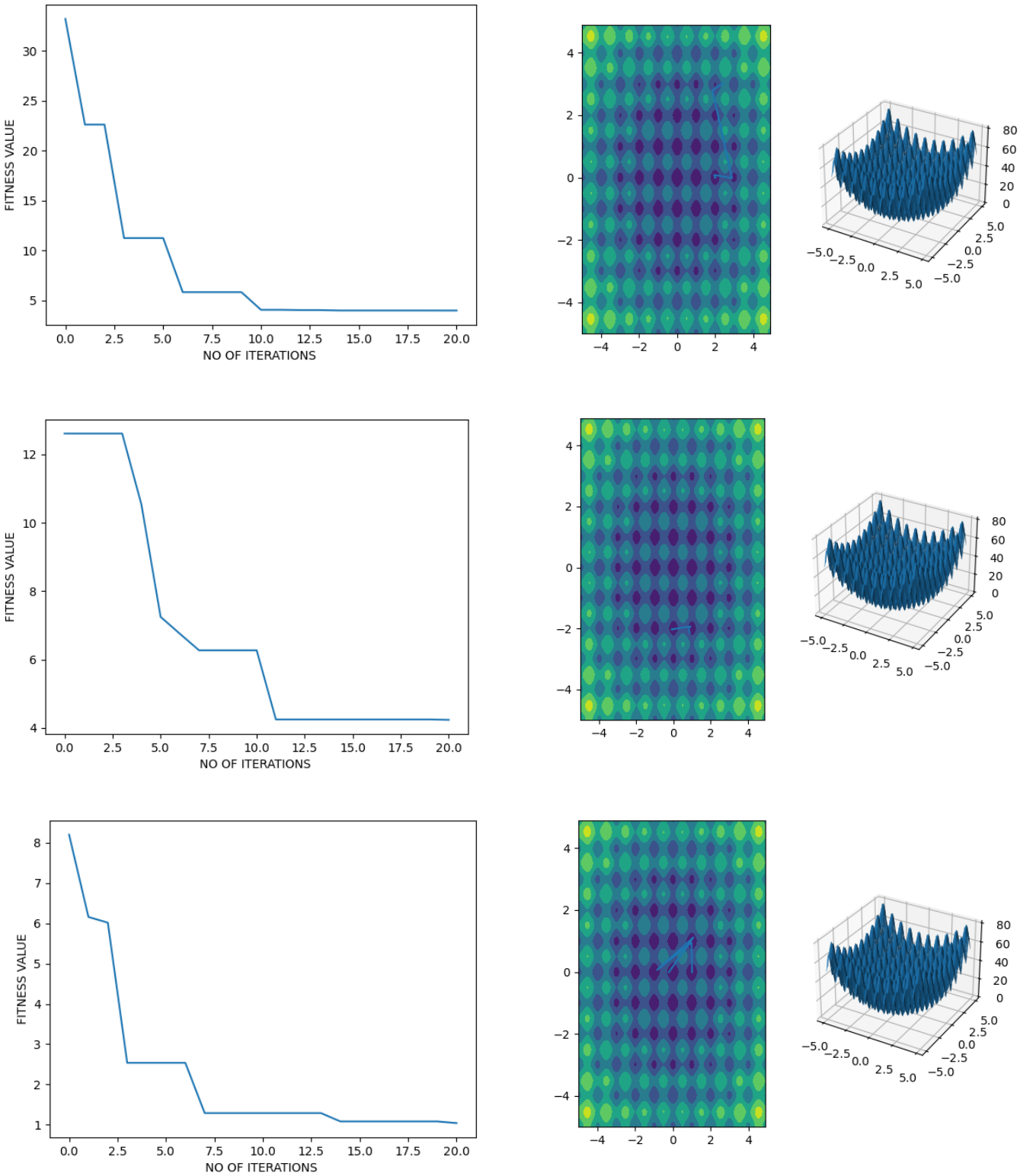


Figure 4: $f(gbest)$ and movement of *gbest*

The above convergence of the particles in each run shows that the three runs converged to different optimal solutions, and all of them did not reach the optimal solution.

To conclude, the PSO algorithm can become trapped into a local optimal solution; hence, it cannot find better solutions because its exploration capability is limited, this problem is common in many optimization algorithms, and it is called Stagnation. However, there are methods to mitigate this problem, such as combining PSO with other optimization algorithms, which make a balance between the exploration and exploitation phases. Approximately all the recent optimization algorithms can solve this problem, but they do not guarantee to reach the same solution in each run due to the stochastic nature of the optimization algorithms. PSO algorithm suffers from partial optimism, which degrades the regulation of its speed and direction.

Also, meta-heuristics such as PSO do not guarantee an optimal solution is ever found.

7 A Variation of PSO - Shuffled Frog Leaping Algorithm

7.1 Introduction

The shuffled frog leping algorithm (SFLA) was introduced by Eusuff and Lansey in 2003 for optimization of water distribution network design. It is a hybrid of PSO and shuffled complex evolution (SCE). SCE is based on the ideo of allowing sub-populations to evolve independently and periodically allowing interactions between sub-populations. SFLA is a nature inspired population based meta-heuristic that imitates the behaviour of a frog population searching for food.

7.2 Basic Concepts

7.2.1 Memetic Evolution

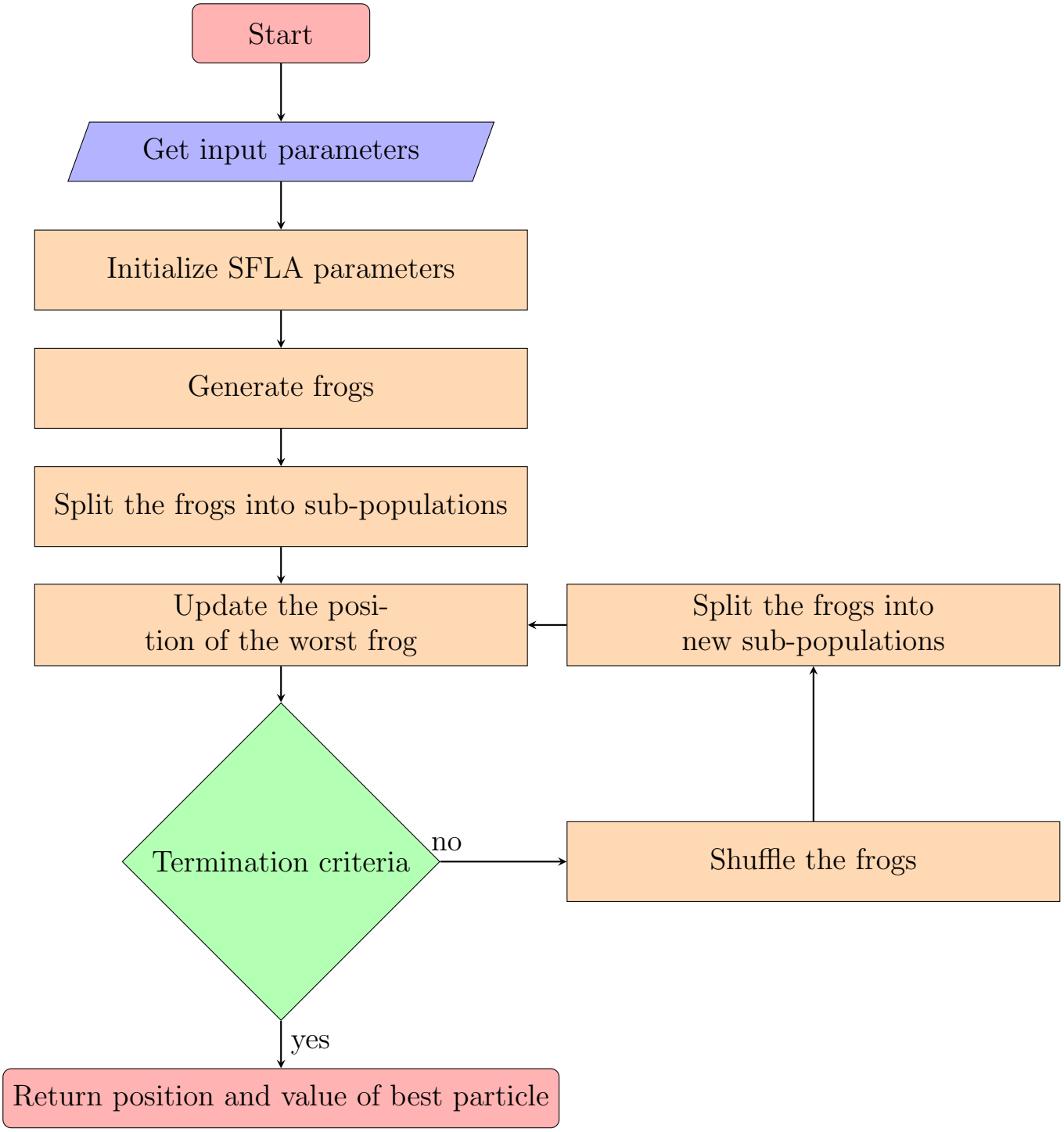
A meme is an idea that spreads from one person to another within a culture. A field of study called memetics arose to explore the concepts and transmission of memes in terms of evolutionary model. Internet memes are an example of this memetic theory.

Genetic algorithm is a popular meta-heuristic that uses genes as unit of evolution to solve optimization problems. SFLA uses memetic evolution as its evolutionary model instead. Memetic and genetic evolution are similar in some ways, i.e., possible solutions are created, selected according to some measure of fitness, combined with other solutions. But memetic evolution differs in many ways: genes are typically transmitted between generations. Usually only fitter parents are taken to the next generation and their children repopulate the generation. But information from one meme can be incorporated in other memes immediately rather than waitng for a full generation. Thus the potential advantage of memetic evolution is information is passed between all individuals in the population rather only parent-children in genetic evolution.

7.2.2 Leaping Frogs

For this algorithm, individual frogs are seen as host for memes. Each meme consists of memetypes. These memetypes represent an idea similar to gene representing a trait in genetic algorithm. Each frog has its location and its fitness value as its memetypes. A lower fitness value means higher fitness for minimization problems. The population consists of sub-populations, called memeplexes. A local search is done for each sub-population. Then the sub-populations are shuffled.

7.2.3 Flowchart for Algorithm



7.3 Algorithm for Minimization Problems

In minimization problems the fitness corresponds to the value of the function optimized. A low function value indicates high fitness. SFLA begins by randomly creating N frogs, (i.e., N solutions). The N frogs are divided into m sub-populations, also called memeplexes. Local search is performed in each sub-population. The local search consists of i_{max} iterations. Each iteration, only the frog's position with worst fitness (highest function value) is updated

as follows:

$$\bar{x}_w \leftarrow \bar{x}_w + r(\bar{x}_b - \bar{x}_w)$$

where $\bar{x}_w$ is the position of frog with worst fitness, r $\in$ [0, 1] is a uniformly distributed random number, and $\bar{x}_b$ is the frog in sub-population with best fitness (least fitness value). However, if fitness does not improve then $\bar{x}_w$ is updated as follows:

$$\bar{x}_w \leftarrow \bar{x}_w + r(\bar{x}_g - \bar{x}_w)$$

where r $\in$ [0, 1] is a new random number and $\bar{x}_g$ is the position of globally best frog of all m sub-populations. If $\bar{x}_w$ is still not improved the $\bar{x}_w$ is updated with a random position.

After local search is done once for each sub-population, the frogs are then shuffled among the sub-populations. Then local search is performed for each sub-population as defined above. This is done for k_{max} iterations.

7.4 Performance for Rastrigin Function

Let N be the number of frogs (individuals), m be the number of sub-population, i be the number of iterations in local search, k be the number of iterations in SFLA algorithm.
The Rastrigin function is defined on 2 - Dimensions as follows:

$$f(x_1, x_2) = 20 + (x_1^2 - 10cos(2\pi x_1)) + (x_2^2 - 10cos(2\pi x_2))$$

The rastrigin function has multiple local minima, hence gradient based methods cannot be used to globally optimize the function. SFLA algorithm works very well for this function. For the domain as x_1, $x_2 \in [-10, 10]$ the global minimum is $f(x) = 0$ at $x_1 = 0$, $x_2 = 0$.
$N = 50$, $m = 5$, $k = 20$, $i = 10$ yields the minimum as $5.0209 \cdot 10^{-10}$ at $x_1 = -1.0492 \cdot 10^{-6}$, $x_2 = -1.1957 \cdot 10^{-6}$ after 1397 function evaluations. This result is very close to actual value and is also reproducable during different runs.

7.5 Optimal Parameter values

From testing various parameters, it has been experimentally found that the number of local search iterations $i = 10$, the number of frogs = 200, the number of sub-population = 20 works well for a large number of optimization problems. However, fine tuning may be required. For the above rastrigin function the number of frogs = 50 and the number of sub-population = 5 also yielded good results but with much faster runtime.

7.6 Disadvantages

As mentioned above, sometimes parameters have to be tweaked to get fast results. As this algorithm has been inspired from nature, mathematical analysis for best input parameters is lacking and must be experimentally determined. Metaheuristics such as SFLA do not guarantee an optimal solution is ever found.

8 References

- Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm by Muzaffar M. Eusuff and Kevin E. Lansey - JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT © ASCE / MAY/JUNE 2003
- Evolutionary Optimization Algorithms by Dan Simon - WILEY publications
- Algorithms for Optimization by Mykel J. Kochenderfer, Tim A. Wheeler - The MIT Press